

Workshop 2010: Workflow Development Demo

Overview: This portion of the workshop will show you how to create an actual workflow. It will use a simple demo workflow to run a “mock” model. You will learn how to run the demo workflow, add new settings using the NED tool and also add additional tasks to enhance the workflow.

1. Create a New Workflow Instance

Log in to the nimbus machine, where <username> is your guest name (e.g. guest1):

```
ssh <username>@nimbus
```

Create a copy of the demo workflow that we will eventually modify:

```
cd /cib/workflow_tool/workflowTesting
cp -R modeldemo <username>demo          (e.g. guest1demo)
```

Examine the configuration file settings for the workflow:

```
cd <username>demo
cat ModelDemoConfig.nif
```

Note the properties at the top of the file, followed by the “ModelSettings” group and “daysToRun” variable.

2. Open the Workflow Configuration from NED

Examine the same configuration from NED. Open up NED if it is closed, by double-clicking on the NEDClient.jar. It should connect to nimbus. Due to the firewall you may be prompted to enter your password twice. There is no passcode to enter.

Once the NED tool is open, let’s load the configuration on nimbus. Go to “File / Open Remote File” on the menu options. You should see the ModelDemoConfig.nif as one of the options. Click “Ok” to load it.

Note the same “ModelSettings” group. You can expand that by clicking on the arrow and you’ll see the “daysToRun” variable. The daysToRun variable is a combo-box or pull-down menu that is used to control the number of timesteps our “mock” model will run. You can change that by double-clicking (Ctrl + click on Mac) on either ModelSettings or daysToRun and changing the value field on the right to a number from 1 to 7.

3. Edit and Test the New Workflow Configuration

Next, we will tell the configuration to run the workflow you just copied. To do that, you will need to switch to NED's "Design Mode". This can be done through "File / Design Mode" in the menu.

Once in Design Mode, right click (Ctrl-click on Mac) on the "Model Demo" item at the top of the tree. Choose "Edit Workflow Properties" in the pop-up menu.

You'll see a list of properties about the workflow. At the top is "Name". You can click on the value of Name and change it to something else, such as "Rob's Model Demo". Press Enter key to set it.

Another important field to note is the "Repository Branch". This tells you where to get your workflow. In this case we are going to set it to the same name as the directory you just created. Set Repository Branch from "modeldemo" to:

<username>demo (e.g. guest1demo)

Press Enter to save the field and then click Ok.

We are ready to submit our new workflow and test it out. Press the "Play" or "Submit" button on the toolbar.

Your window will change to a tree view showing the running tasks. This is a simple demo so there isn't a lot happening. You'll see an "Init Model" task that should be green and turn yellow and a "Loop" task that is blue, waiting to run.

The arrow beside "Loop" indicates that it contains other tasks. Click the arrow to show the "Run Model" task. This task will loop the number of times that you set "daysToRun". You'll see it say Run 2 of 3, for example on the second loop.

To follow along with the task output as it runs, right-click (Ctrl-click on Mac) on the "Run Model" task and select "View Task Log" in the pop-up. You should see it telling you that it's running the model each time it loops.

4. Add a New Model Option to the Configuration

Now that we are familiar with the workflow, let's add an enhancement: we'd like the workflow to notify us via email when it finishes. So we need to add: an email address variable and also the task that actually sends the email.

First, let's add the email address to our configuration. We are already in Design Mode so right-click (Ctrl-click on Mac) on "ModelSettings". Select "Add A Variable" in the pop-up.

In the variable dialog, set the following items:

Name: emailAddress

Description: Email to notify when workflow completes

Values: <your email address> (e.g. Robert.W.Burns@nasa.gov)

This variable will now be made available in our workflow when it runs. All that's left is to add an email task so that we actually send an email.

5. Add an “Email User” Task to the Workflow

Switch back to your Terminal window. Let's create the email notification. In the terminal window, use your favorite editor (e.g. gedit)

```
gedit emailUser.sh
```

In the editor type the following lines:

```
source $NED_WORKING_DIR/ModelVariables.bash
mail -s "$NED_UNIQUE_ID done" $emailAddress < /dev/null
```

Save that and exit the editor. Now we've created a script to notify the user, but the workflow must understand how to run it. Let's take a look at how the tasks are defined in the workflow. Once again, use your favorite editor to modify the workflowArchitecture.wif file. This file defines the tasks that make up the workflow.

```
gedit workflowArchitecture.wif
```

You'll notice several five sections in the file, which should look somewhat familiar: Workflow Architecture at the top has various properties about your workflow. Underneath are the tasks you saw running in the workflow: Init Model, Loop, and Run Model.

The very last text block is our email task, pre-added for convenience. It is currently commented out with “#” in front of the lines. Delete all six “#” characters at the front of each line. Note that this task is told to run the “emailUser.sh” script identified by the “EXECUTABLE_OBJECT” field.

Save that and exit the editor.

Now switch back to NED so we can test our modified workflow. Click on the “Submit” button (play icon) once again. This time when the view switches, you should see a new task appear at the bottom. If everything works correctly, there should be an email sent to the address you entered.

You've now created and modified a workflow.